09/842,527   MS158543.01/MSFTP194US

<u>REMARKS</u>

Claims 1-49 are currently pending in the subject application and are presently under consideration. Claims 41 and 42 have been amended as shown at pages 9 of the Reply.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I.   **<u>Rejection of Claims 36-39, 43-46, 48, and 49 Under 35 U.S.C. §101</u>**

Claims 36-39, 43-46, 48, and 49 stand rejected as being non-statutory because they are directed to a system without recitation of a computer or a computer-readable medium embodying the claimed program elements. It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. The Federal Circuit has clearly established in *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005) and *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed.Cir. 1999) that inventions such as that claimed by applicant's is statutory.

> This court must also decide whether software code made in the United States and exported abroad is a "component of a patented invention" under 271(f)... Section 271(f) refers to "components of a patented invention."... Title 35, section 101, explains that an invention includes "any new and useful process, machine, manufacture or composition of matter."... Without question, *software code alone qualifies as an invention eligible for patenting under these categories*, at least as processes. *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005). (Emphasis added).

The Federal Circuit in *Eolas Techs., Inc. v. Microsoft Corp.* clearly established that software code alone is statutory subject matter. Independent claims 36, 43 and 48 recite a *computer implemented system*. A system by itself is statutory subject matter. By the standards set forth in the above decision, a computer implemented system in the form of software, hardware, or the combination of both clearly falls within the categories of statutory subject matter.

12

Furthermore, the subject claims produce a useful, concrete, and tangible result.

> Because the claimed process [method] applies the Boolean principle to produce a useful, concrete, tangible result ... on its face the claimed process comfortably falls within the scope of §101. *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed.Cir. 1999); *See State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1601 (Fed.Cir.1998) (finding a system implementing a financial management structure satisfied §101 because it constituted a practical application of a mathematical algorithm by producing a useful, concrete and tangible result).

As provided above, the legal standard set forth by the Federal Circuit in *&T Corp. v. Excel Communications, Inc* for determining whether a claim is directed towards statutory subject matter is whether a claim can be applied in a practical application to produce a useful, concrete, and tangible result. The subject claims disclose a code generator that can convert a language neutral representation of code into a high level language code which is a concrete, useful, and tangible result. The high level language code can be used to document the code in a more user readable manner or to edit and compile the code into a form appropriate for alternate computer operating systems.

In view of at least the foregoing, it is readily apparent that applicants' invention as recited in independent claims 36, 43, and 48 (and associated dependent claims 37-39, 44-46 and 49) is statutory subject matter and produces a useful, concrete, and tangible result. Withdrawal of this rejection is requested.

II.   Rejection of Claims 1-49 Under 35 U.S.C. §103(a)

Claims 1-49 are rejected under 35 U.S.C. §103(a) as being unpatentable over Gustafsson *et al.* (US 6,067,413) in view of Dyer ("Java Decompiles compared," Java World, 7/1997). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Gustafsson *et al.* in view of Dyer fails to teach or suggest each and every limitation of applicants' claimed invention.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of

13

obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The subject invention relates to creation of a coding language neutral representation of a compile unit from language specific code. For instance, the applicants' invention can generate a language neutral representation of a compile unit from a representation of the compile unit in high level language code, such as C#. The language neutral representation can then be converted to any of a plurality of representations of the compile unit in various high or low level coding languages, such as C, C++, or Assembly. In particular, as recited in amended independent claim 1 (and similarly recited in independent claims 14, 23, 32, 36, 40, 43 and 47-48), the applicants' claimed invention can *convert the language neutral representation of the compile unit to a corresponding representation of the compile unit in at least one high-level language code.*

Contrary to assertions in the Office Action, Gustafsson *et al.* in view of Dyer, fails to teach or suggest the aforementioned novel aspects of applicants' invention as recited in the subject claims. Gustafsson *et al.* as evidence of prior art disclosing a language neutral representation of a compile unit. However, the cited are discloses compiling a plurality of high level languages into a common runtime representation. Specifically, the common representation is disclosed as the compiled version of one of the high level languages. Gustafsson *et al.* discloses that the compilers for all of the high level languages must be modified to produce output representative of the compiled output of a selected high level language. Therefore, the common representation is not language neutral. Rather, it is specific to one of the high level languages and the compilers for the other high level languages are modified to produce it.

Moreover, Dyer is cited as evidence of prior art disclosing converting a language neutral representation into at least one high level language code. However, the cited art discusses examples of Java decompilers. Specifically, conversion of Java class files to Java source code is

14

disclosed. Java class files are not language neutral representations. They are specific to Java source code. Therefore, Dyer does not disclose converting a language neutral representation of a compile unit into at least one high level language.

Additionally, independent claim 41 recites *mapping each of a plurality of programmatic constructs of the first high level language code to a corresponding class of a plurality of language-neutral classes, ... arranging the objects ... to define the language neutral representation; converting the language-neutral representation of the compile unit into a corresponding representation in at least one second high-level language code, the second high level language code is in a different high level language than the first high level language code.* Gustafsson *et al.* discloses compiling a first high level language code into compiled code representative of a second high level language. However, the cited reference does not suggest converting the language specific compiled code into the second high level language code. Dyer, only discusses a single high level language, Java, and conversion of language specific Java class files into Java source code. Gustafsson *et al.* and Dyer fail to disclose a language neutral representation of a compile unit and also fail to teach or suggest converting a language neutral representation of a compile unit into a high level language code. Therefore, the cited references also do not teach or suggest converting a first high level language code into a language neutral representation and then converting the language neutral representation into a second high level language code that is in a different high level language than the first.

Accordingly, applicants' representative respectfully submits that Gustafsson *et al.* and Dyer, alone or in combination, fail to teach or suggest all limitations of applicants' invention as recited in independent claims 1, 14, 23, 32, 36, 40, 41, 43 and 47-48 (and all claims that depend respectively therefrom), and thus fails to make obvious the claimed invention. This rejection should be withdrawn.
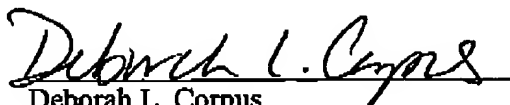
CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP194US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN & TUROCY, LLP

Deborah L. Corpus
Reg. No. 47,753

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731

16